

# 다중 패치로 구성된 포인트 클라우드 데이터 압축 파일포맷

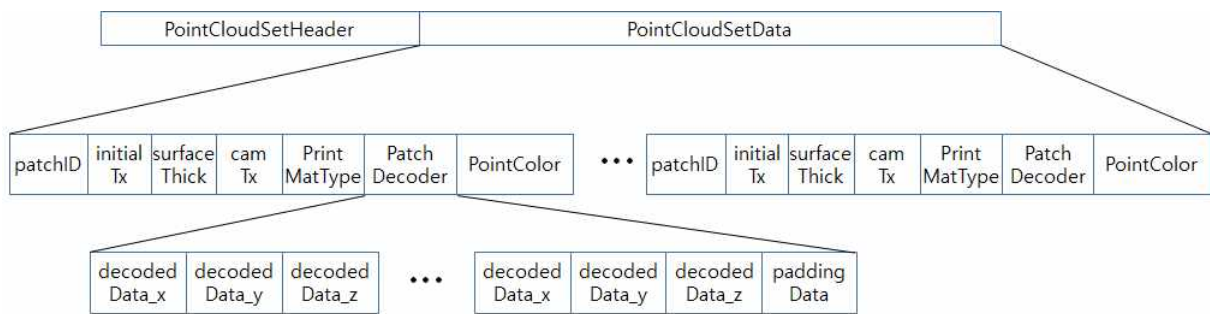
이승욱 한국전자통신연구원 책임연구원

## 1. 머리말

본고는 스캐너로 생성한 포인트 클라우드 데이터를 저장할 때 사용되는 비트스트림의 구조, 헤더정보, 데이터 정보 및 관련된 선택스와 시맨틱스를 기술함으로써 다중패치를 압축하여 저장할 수 있는 파일포맷을 소개한다.

## 2. 포인트 클라우드 파일의 비트스트림 구조

일반적인 비트스트림 형태와 같이 헤더(PointCloudHeader) 및 데이터(PointCloudData)로 구성되며, 데이터는 패치 개수만큼의 정보가 반복된다. 헤더에는 비트스트림을 구성하는 일반적 내용이 기술되며, 데이터 부분에 실제 패치로 구성된 포인트 클라우드 정보가 기술된다.



[그림 1] 비트스트림 구조

## 3. 비트스트림의 선택스 및 시맨틱스

바이너리 기반의 포인트 클라우드 파일포맷의 선택스 및 시맨틱스를 정의한다.

### 3.1 PointCloudSet

포인트 클라우드 데이터 파일포맷의 최상위 노드정보이다.

#### 3.1.1 선택스

```

class PointCloudSet {
    PointCloudSetHeader    header;
    PointCloudSetData      data;
}

```

### 3.1.2 시맨틱스

- PointCloudSet: 바이너리 기반 포인트 클라우드 데이터의 최상위 노드
- header: PointCloudSet의 헤더정보
- data: PointCloudSet의 실제 데이터 정보

## 3.2 PointCloudSetHeader

### 3.2.1 선택스

```

class PointCloudSetHeader {
    unsigned int(32)    streamSize;
    unsigned char(1024)    copyRight;
    unsigned char(1024)    acquisitionAlgo;
    bit(8)    unit;
    float(32)    minimumVerticalResolution;

    unsigned int(32)    numOfPatch;
    if (numOfPatch>0)
    {
        for (int i = 0; i < numOfPatch; i++)
        {
            unsigned int(32)    numOfPoint[i];
            unsigned int(32)    numOfColor[i];

            if (numOfPoint[i] > 0)
            222{
                for (j = 0; j<3; i++) {
                    float(32)    quantMinPoint[i];
                    float(32)    quantRangePoint[i];
                }
            }
        }

        bit(8)    QPforPoint;
    }
}

```

### 3.2.2 시맨틱스

- streamSize: PointCloudSet의 전체 파일 크기를 바이트 단위로 나타낸다.
- copyRight: 스트링 형태로 정의된 저작권 정보를 나타낸다. 3D 스캐너의 경우 데이터만 있

으면 제품을 생산 할 수 있기에 저작권의 정보가 중요하다.

- acquisitionAlgo: 스트링 형태로 정의된 포인트 클라우드 생성 알고리즘을 정의한다. 레이저 방식, 패턴 방식, 스테레오 방식 등 다양한 부가 정보를 삽입할 수 있다.
- unit: 8비트의 정보로 실제 사용되는 포인트의 3차원 좌표에 정의된 값의 단위(unit)를 정의한다. 만약 이 값이 mm로 정의되면, 모든 좌표는 mm로 해석된다.
- minimumVerticalResolution: 3D 모델 획득시의 해상도를 나타낸다. 디자이너에 의해 생성된 모델이 아닌 스캐너를 통해 만들어진 모델의 경우에서 사용되는 스캔 해상도를 의미한다. 이 값이 0.1로 설정된 경우 이 값보다 작은 해상도의 출력 해상도(y-축 적층 해상도)는 의미가 없다. 즉, 이 값은 수직방면의 최소 출력 해상도를 나타낸다.
- numOfPatch: 패치의 개수를 정의한다. 만약 이 값이 1이면 정합된 패치 혹은 한 번에 스캔된 모델을 의미하고, 이 값이 2 이상이면, 패치 정합 과정을 통해 하나의 포인트 클라우드로 정렬하는 작업이 필요하다.
- numOfPoint: 각 패치에 존재하는 포인트의 개수.
- numOfColor: 각 패치에 존재하는 포인트 컬러의 개수, 이 값은 numOfPoint와 같은 값이거나 '0'이다. 컬러를 지원하지 않는 포인트 클라우드 데이터도 존재하기 때문이다.
- quantMinPoint : 양자화 과정에서 사용되는 포인트의 최소 값을 정의한다.
- quantRangePoint : 양자화 과정에서 사용되는 범위 값을 정의한다.

<표 1> unit에 대한 시멘틱 정보

값	정보
0	millimeter
1	centimeter
2	meter
3	inch
4	feet
5	yard
나머지	reserved

### 3.3 PointCloutSetData

#### 3.3.1 선택스

```

class PointCloudSetData {
    if (numOfPatch>0) {
        for (int i = 0; i < numOfPatch; i++)
        {
            unsigned int(32) patchID;
            for (int j = 0; j < 12; j++)
            {
                float(32) initialTransform[j];
            }
        }
    }
}

```



### 3.4 EachPatchDecoder

#### 3.4.1 신택스

```
class EachPatchDecoder(numData, dim, QP) {  
    for (int i = 0; i<numData; i++) {  
        for (int j = 0; j<dim; j++) {  
            bit(QP) decodedData[j];  
        }  
    }  
    unsigned int (32) numPaddingBits;  
    bit(numPaddingBits) paddingData;  
}
```

#### 3.4.2 시맨틱스

- decodedData : 각 x, y, z 좌표계 상에서의 디코딩된 포인트 값을 정의한다.

역양자화 및 차분 코딩을 이용하여 복원한다.

즉, 첫 번째 디코딩 되는 경우는 역양자화만 이용하고 두 번째 부터는 역양자화를 수행한 후 이전 값을 더한다. 첫 번째 복원되는 값인지 여부에 따라 다음과 같이 계산된다.

case 1. 첫번째 값인 경우:

$$V_r = [v_q / (2^{QP} - 1)] * quantRange + quantMin$$

case 2. 첫 번째 이후의 경우

$$V_r = V_{r\_pre} + [v_q / (2^{QP} - 1)] * quantRange + quantMin$$

$V_r$ : 역 양자화 과정을 거친 최종 결과 값

$v_q$ : 양자화 된 입력 값

quantRange: 각 x, y, z축의 양자화 범위 값, 헤더 정보에 있는 quantRangePoint 값

quantMin: 각 x, y, z축의 최소값, 헤더 정보에 있는 quantMinPoint 값

$V_{r\_pre}$ : 이전의 디코딩된 값

- paddingData: numPaddingBits 비트만큼의 크기를 가지는 패딩 데이터로 8-bit alignment 를 위한 패딩비트로 사용되며, 추출된 정보는 사용하지 않고 버린다.